REMARKS

This paper is responsive to the Final Office Action dated April 7, 2006. All rejections are respectfully traversed. Reconsideration of all claims is respectfully requested.

At paragraph 1 of the Office Action, the Examiner rejected claims 1-4, 9-10, 12-20, 25-26, 28-37, 39, and 41-47 under 35 U.S.C. 103(a), citing the combination of "XNAMI - An eXtensible XML-based paradigm for Network and Application Management Instrumentation" by John et al. ("John et al.") with U.S. Patent Number 5,541,911 of Nilakantan et al. ("Nilakantan et al."). Applicants respectfully traverse this rejection.

Applicants respectfully urge that a fundamental difference exists between the present independent claims 1, 17 and 33 and the approaches of John et al. and Nilakantan et al., in that the present independent claims 1, 17 and 33 provide ***a description of a data forwarding service by specification of a class of objects for the data forwarding service in a markup language document, and instantiate and launch the data forwarding service on a network device based on the class of objects from the markup language document***, whereas John et al. and Nilakantan et al. describe techniques for defining and/or manipulating MIB table variables in order to control services on a network device that operate in response to the MIB. This distinction is set forth in further detail below.

John et al. discloses a system in which XML is used to convey *a new MIB variable definition,* together with *code operable to process subsequent GET and SET requests for that MIB variable*. This teaching of John et al. is found in the SNMP PDU containing the "XML string for variable definition" and "Compressed Method bytecode" shown in Fig. 8 of John et al.

on page 7. With regard to the "XML string for variable definition" shown in Fig. 8, John et al. state as follows beginning at line 32 of the left hand column of page 7:

> The value to which the manager SETs mib_proxy is a string containing ***a description in XML of the new objects in the subtree***. (emphasis added)

John et al. go on to describe the "Compressed Method bytecode" in the SNMP PDU of Fig. 8 as follows beginning at line 3 of the right hand column of page 7:

> The values which the manager passes to the methods_proxy object are strings containing the compressed Java byte code, one for each leaf node being added to the tree. The bytecode strings contain the code for doing GET and SET operations on the newly-added MIB objects. As the new MIB objects are created, ***the bytecode for each node is decompressed and loaded as a Java class containing two methods, one for a GET on the MIB object and one for a SET***. (emphasis added)

The above sections of John et al. teach that the Java class defined by the SNMP PDU of Fig. 8 indicates *methods for GET and SET operations on newly created MIB variable*. As described in Section 2.2 of John et al., SNMP SET operations are modification operations on MIB variables, while SNMP GET operations are a type of retrieval operations on MIB variables.

Nilakantan et al. teaches filtering operation based on MIB variables. Fig. 3 of Nilakantan et al. shows a Smart Filter master and a Smart Filter agent. As described in columns 5 and 6 of Nilakantan et al., Smart Filter master code is implemented within a central router, which includes an interface to an SNMP transport mechanism that is coupled to a port for communication with the Smart Filter agent over a network. The Smart Filter agent of Nilakantan et al. "includes an SNMP management information base MIB." The Smart Filter agent performs packet spoofing and packet filtering for a LAN in response to information in the MIB.

In view of the above, the combination John et al. and Nilakantan et al. results in a system that uses the techniques in John et al. for conveying *a new MIB variable definition,* together with *code operable to process subsequent GET and SET requests for that MIB variable*, to another device storing a MIB used by the Smart Filter of Nilakantan et al. to perform packet spoofing and packet filtering for a LAN. Nothing in the combination of John et al. and Nilakantan et al. provides any hint or suggestion of even the desirability of controlling a data forwarding service in a network device by:

> receiving at the network device a document written in accordance with a markup language and a corresponding document definition, ***wherein the document describes the data forwarding service by specifying a class of objects for the data forwarding service***;
>
> . . .
>
> executing the data forwarding service on the network device . . . wherein the executing includes ***instantiating and launching the data forwarding service in the data forwarding device based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device to filter network traffic***. (emphasis added)

as in the present independent claim 1 (analogous features are present in independent claims 17 and 33). The only mark-up language described in the combination of John et al. and Nilakantan et al. is the XML document describing the MIB (page 121, column 1, lines 1-8), and the XML string for variable definition shown in Fig. 8 of John et al. Neither of these descriptions includes any teaching of a markup language document that *describes a data forwarding service by specifying a class of objects for the data forwarding service*, nor any *instantiating and launching of the data forwarding service based on the class of objects in the data forwarding service*, as in the present independent claims 1, 17 and 33. As a result, the combination of John et al. and Nilakantan et al. fails to allow dynamic instantiation and launching of a service based on an object class specifying the service contained in a markup language document, which is an advantage of the present independent claims 1, 17 and 33 over such systems. In contrast, John et al. and Nilakantan et al. are limited to providing GET and SET functions and definitions for MIB variables, and then using the resulting MIB to control the Smart Filter Agent of Nilakantan et al.

For these reasons, Applicants respectfully urge that the combination of John et al. and Nilakantan et al. does not disclose or suggest all the limitations of the present independent claims 1, 17 and 33. John et al. and Nilakantan et al. accordingly do not form a *prima facie* case of obviousness with regard to the present independent claims 1, 17 and 33 under 35 U.S.C. 103. As to dependent claims 2-4, 9-10, 12-16, 18-20, 25-26, 28-32, 34-37, 39, 41, and 44-47, they each depend from claims 1, 17 and 33, and are respectfully believed to be patentable over the combination of John et al. and Nilakantan et al. for at least the same reasons.

At paragraph 2 of the Office Action, the Examiner rejected claims 5-8, 11, 21-24, 27, 38 and 48-50 for obviousness under 35 U.S.C. 103(a), again citing John et al. and Nilakantan et al., in further combination with "An Introduction to the Extensible Markup Language XML" by Bryan ("Bryan"). Applicants respectfully traverse this rejection.

As with John et al. and Nilakantan et al., nowhere in Bryan is there disclosed or suggested any method or system for controlling a data forwarding service in a network device comprising a data forwarding device, as in the present independent claims 1, 17, 33 and 48, from which claims 5-8, 11, 21-24, 27, 38 and 49-50 depend. Neither John et al., Nilakantan et al., nor Bryan teach or suggest:

> receiving at the network device a document written in accordance with a markup language and a corresponding document definition, ***wherein the document describes the data forwarding service by specifying a class of objects for the data forwarding service***;
>
> . . .

> executing the data forwarding service on the network device . . . wherein the executing includes ***instantiating and launching the data forwarding service in the data forwarding device based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device to filter network traffic***. (emphasis added)

Bryan provides an overview of commonly used components in XML, including Document Type Definitions (DTDs) to formally identify the relationships between elements in an XML document. Neither John et al., Nilakantan et al. nor Bryan include even a suggestion of the desirability of the above indicated features of the present independent claims, from which claims 5-8, 11, 21-24, 27, 38 and 49-50 depend.

Applicants therefore respectfully urge that the combination of John et al., Nilakantan et al., and Bryan does not disclose or suggest all the limitations of the present independent claims 1, 17, 33 and 48. The combination of John et al., Nilakantan et al. and Bryan accordingly does not support a *prima facie* case of obviousness under 35 U.S.C. 103 with regard to the present independent claims 1, 17, 33 and 48. As to dependent claims 5-8, 11, 21-24, 27, 38 and 49-50, they each depend from claims 1, 17, 33 and 48, and are respectfully believed to be patentable over the combination of John et al., Nilakantan et al. and Bryan for at least the same reasons.

At paragraph 3 of the Office Action, the Examiner rejected dependent claims 40 and 46 for obviousness under 35 U.S.C. 103(a), again citing John et al. and Nilakantan et al., as well as lines 8-16 on page 9 of the Applicants' Specification, and "Dynamic Classification in Silicon-based Forwarding Engine Environments" (Jaeger). Applicants respectfully traverse this rejection.

Applicants again respectfully note that the statements in lines 8-16 on page 9 of the Specification do not constitute an admission of prior art, at least in part because the first words of that paragraph are "In one example of the invention. . . " Additionally, the section of the Specification containing those lines is entitled "DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS". Moreover, the cited text repeatedly refers to the ORE^{TM} embodiment as "an example of the invention". For at least these reasons the statements in lines 8-16 on page 9 of the Specification should not be considered an admission of prior art.

As with John et al. and Nilakantan et al., nowhere in Jaeger is there disclosed or suggested any method or system for controlling a data forwarding service in a network device comprising a data forwarding device as in the present independent claim 33, from which claims 40 and 46 depend. Neither John et al., Nilakantan et al., nor Jaeger teach or suggest a network device for locally performing a data forwarding service in accordance with a received document written in a document markup language, wherein the network device comprises a data forwarding device, including:

> a parser that is adapted to parse the received document in accordance with a document definition to obtain an identifier of the service, ***wherein the parsing determines at least one parameter describing the data forwarding service by specifying a class of objects for the data forwarding service***; and
>
> a service launcher that is adapted to launch the data forwarding service corresponding to the identifier parsed from the received document, ***wherein the service launcher instantiates and launches the data forwarding service in the data forwarding device upon completion of the parsing based on the class of objects for the data forwarding service, and wherein the data forwarding service configures a forwarding architecture in the network device operable to filter network traffic***. (emphasis added)

Jaeger discloses a programmable network architecture built on a Gigabit Ethernet L3 Routing switch to support downloadable services. Neither John et al., Nilakantan et al. nor Jaeger include even a suggestion of the desirability of the above indicated features of the present independent claim 33, from which claims 40 and 46 depend. For these reasons, Applicants respectfully urge that the combination of John et al. and Nilakantan et al. with Jaeger does not disclose or suggest all the limitations of the present independent claim 33, from which dependent claims 40 and 46 depend.

Applicants further submit that the Examiner has not established a sufficient motivation to combine John et al. and Nilakantan et al. with Jaeger. A *prima facie* case of obviousness under 35 U.S.C. 103 must include a showing of a suggestion, teaching or motivation that would have led a person of ordinary skill in the art to combine the cited references *in the particular manner claimed*. See In re Dembiczak, 175 F.3d 994, 998 (Fed. Cir. 1999), and In re Kotzab, 217 F.3d 1365, 1371 (Fed. Cir. 2000). "[C]ombining prior art references without evidence of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability—the essence of hindsight." Dembiczak, 175 F.3d at 999. In the present rejection, the Examiner again asserts that a skilled person would be motivated to combine John et al. with Jaeger based on teachings in the Conclusion of Jaeger, which states that ORE "supports the creation of services in Java that are extensible, portable, and easily distributed over the network". Applicants continue to respectfully disagree, and note that the solution described in John et al. concerns providing *definitions of and access to new MIB variables*, and does not point to any need for "creation of services in Java". Accordingly, Applicants respectfully urge that one skilled in the art would not be motivated to modify John et al. to include the teachings of Jaeger et al. for the reasons cited by the Examiner, since the systems of John et al. and Jaeger have different objectives in this regard. Moreover, the Examiner has provided no motivation whatsoever for combining Nilakantan et al. with Jaeger.

The combination of John et al. and Nilakantan et al. with Jaeger accordingly does not support a *prima facie* case of obviousness under 35 U.S.C. 103 with regard to the present independent claim 33, and dependent claims 40 and 46 are respectfully believed to be patentable over the combination of John et al. and Nilakantan et al. with Jaeger for at least the same reasons.

For the above reasons, and in view of the amendments to the claims herein, Applicants respectfully urge that the present claims are allowable over the prior art of record, and respectfully request that the Examiner's rejections be withdrawn. This application is now considered to be in condition for allowance and such action is earnestly solicited.

Applicants have made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone David A. Dagg, Applicants' Attorney at 617-630-1131 so that such issues may be resolved as expeditiously as possible.

Respectfully Submitted,

June 7, 2006
Date

/David Dagg/
David A. Dagg, Reg. No. 37,809
Attorney/Agent for Applicant(s)
McGuinness & Manaras LLP
125 Nagog Park Drive
Acton, MA 01720
(617) 630-1131